Performing Text-driven Robot Grasp Tasks with CLIP-NeRF Approach

Jiahao Ren*
Brown University
Providence, USA
jiahao_ren@brown.edu

Yumeng Ma*
Brown University
Providence, USA
yumeng_ma1@brown.edu

ABSTRACT

We present a novel approach to scene modeling for text-guided robot grasping tasks. Our model, CLIP-feature field, is trained using Contrastive-Language-Image Pretraining (CLIP) features from multiple viewpoints. It can learn to map spatial locations to semantic embedding vectors, enabling robots to understand the semantic meaning of objects in their surroundings. Unlike traditional scene models, our approach requires minimal supervision and labeling from users and can capture objects of arbitrary sizes. We demonstrate that objects can be detected using tokens that describe their semantic meaning, color, and other characteristics. We show that our CLIP-feature field can guide robots in performing grasp tasks under text-based instructions. By leveraging the continuous and implicit nature of our scene model, we enable robots to accurately and efficiently locate and grasp objects in complex environments. Our approach has the potential to significantly improve the capabilities of text-guided robotic systems, making them more adaptable and flexible in real-world scenarios.

INTRODUCTION

Robotics researchers have long been interested in the topic of robotic grasping, as it is a fundamental but complex skill for robots to master. Successfully grasping objects is a significant challenge for robotic systems as it requires the coordination of perception, planning, and control. However, current solutions for robotic grasping still lag behind human performance [6, 10], particularly when faced with unstructured and unpredictable environments.

Neural implicit representations or neural fields (NeRF) have shown to be a promising approach for representing a variety of signals. These neural radiance fields, and implicit neural representations more generally, can serve as differentiable databases of spatio-temporal state that can be used by robots for scene understanding, SLAM, and planning [9]. New research indicates that large-scale vision-language models trained with weak supervision on web-scale data are capable of capturing significant semantic abstractions [2]. However, these models are restricted to processing a single 2D image as input, which limits their applications. Therefore, the challenge remains of finding the most effective way to leverage these models to facilitate 3D reasoning and take advantage of their capabilities in this domain.

In this work, we introduce a CLIP-NeRF method for constructing weakly supervised semantic neural fields. These fields are trained using a contrastive loss function that penalizes discrepancies between the estimated and synthesized ground truth CLIP features at specific locations in 3D space. We present qualitative examples of image view localization, in which we successfully localize images based on accompanying text descriptions within the space. Furthermore, we demonstrate the potential for robots to utilize our CLIP-feature field to execute intricate grip-related tasks with precision.

RELATED WORKS

Previous studies have shown that features from weakly-supervised web-image trained models, such as CLIP, can be utilized for robotic scene understanding [8, 11]. Among these studies, [2] uses CLIP embeddings to label points in 3D space based on a single view, similar to our work. However, their approach involves a two-step process, whereas our approach, which employs an implicit map trained with CLIP embeddings, directly generates a semantic vector for each point in space.

Another related study is Language Embedded Radiance Fields (LERF) [3], which proposes a method for generating dense 3D representations of objects from a single image using a neural network that embeds natural language queries. LERF's approach allows for more intuitive object manipulations by robots. Our approach draws inspiration from this but leverages CLIP embeddings to train an implicit map for generating semantic vectors in 3D space. By doing so, we demonstrated the potential of weakly-supervised models using CLIP, to enhance robotic grasping.

TECHNICAL APPROACH

We aim to enable robots to understand the semantic meaning of objects in their surroundings and perform grasp tasks under text-based instructions using minimal supervision and labeling from users. Our work mainly consists of two parts: (1) a CLIP-feature encoder that encodes RGB frames with per-pixel CLIP features, and (2) a CLIP-feature field model that predicts the CLIP feature values and density at any given coordinate, which is trained using RGB frames and per-pixel CLIP feature maps obtained by (1). The workflow of our model is illustrated

CSCI2951-K Topics in Collaborative Robotics Final Report, Spring 2023

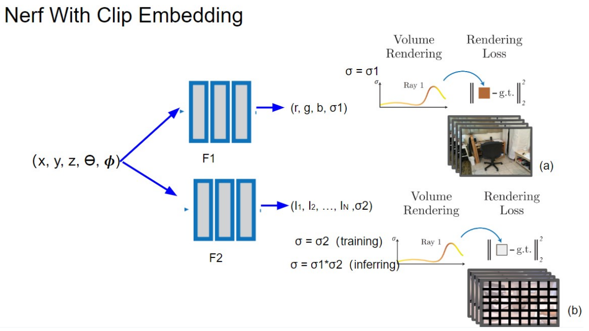

Figure 1. The model takes in two sets of data to train: (a) RGB frames of the scene we intend to reconstruct, and (b) CLIP-feature maps of the RGB frames Using these two sets of data as reference, the model takes a coordinate in 3D space and a view direction as input, and tries to predict the RGB value along with their opacity 1, as well as the clip feature value along with their opacity 2. The model uses volumetric rendering to accumulate the RGB and CLIP-feature values along a given ray, and sample the ground truth RGB and CLIPfeatures from (a) and (b).

in Figure 1. This approach was then demonstrated on a Franka Panda manipulator.

CLIP Feature Encoder

CLIP is a powerful neural network trained on a diverse set of (image, text) pairs, enabling it to predict the most relevant text snippet for a given image based on natural language instructions [7]. Its ability to encode both images and text into embeddings in a joint space allows for the calculation of cosine similarity between the two embeddings, indicating their level of relatedness. However, CLIP's image-level embedding lacks spatial resolution, which limits its ability to capture fine object details and therefore makes it unsuitable for reconstruction tasks.

To overcome this limitation, we drew inspiration from previous approaches that extract dense features [5] or relevancy maps [13] to encode smaller regions of the image. Initially, we attempted to divide the image using superpixels and extract feature maps, but we found that using a sliding window to uniformly sample from the image produced more stable and detailed results.

Our proposed method, as shown in Figure 2, generates a CLIP embedding for any given coordinate on the image. We sample the area around the point of interest to obtain a smaller image, which is then encoded by the CLIP image encoder to output a CLIP feature describing the objects present in the current window. By adjusting the window size appropriately, the resulting feature captures the semantic meaning of the image at the current coordinate, enabling us to reconstruct fine object details.

CLIP-Feature Field

The CLIP-feature field is an important component of our approach and shares the same structure as NeRF. While NeRF outputs an RGB value and corresponding density $\sigma 1$ given a 3D coordinate and viewing direction, the CLIP field outputs a CLIP feature and corresponding density $\sigma 2$. By using volumetric rendering along a ray, we obtain the final CLIP feature and optimize the neural network using a previously encoded feature map.

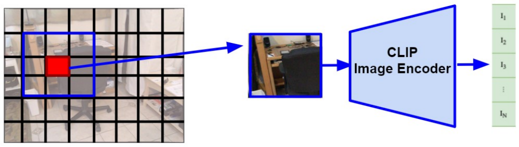

Figure 2. The diagram shows how the per-pixel CLIP-feature map is encoded: generate a CLIP feature for an arbitrary coordinate on the RGB image and sample the image area around it using some fixed window size. We then use CLIP to encode the image within this window to get an embedding. in space. One notable consequence is that our approach integrates semantic information from multiple views into spatial memory.

However, the CLIP model's inherent ambiguity means that the obtained features may not always accurately represent the semantic meaning at a given coordinate. While the features tend to be more accurate at the center of objects due to the window being able to capture their shape, they become less accurate on the object's surface and the surrounding space. In addition, the CLIP feature is not multi-view consistent, resulting in blurred features at the boundary between objects and surrounding space. To address this issue, we make use of the RGB density $\sigma 1$ obtained from NeRF to guide the rendering process. By using $\sigma 1$ to penalize the density in empty spaces, we can mark the clear boundary between objects and empty spaces, as color changes occur in high frequency on object edges. Moreover, a low $\sigma 1$ value indicates a low material density, and so the corresponding CLIP feature density $\sigma 2$ should be low as well. We train NeRF and the CLIP feature field separately during the training phase to ensure that the density queried in NeRF is not affected by ambiguous CLIP encoding. In the inference phase, we use $\sigma = \sigma 1 * \sigma 2$ as the opacity when rendering the CLIP feature along a ray, effectively ignoring high CLIP density in empty spaces and leading to better details around object surfaces.

EVALUATION

Our research mainly focused on conducting qualitative experiments to gain a better understanding of our model's behavior. We explored different window sizes, novel view synthesis, and object grounding by text. To reduce computational complexity, we employed a sliding window technique, moving the window in steps of 8 pixels to down-sample the feature map. This approach did not affect the reconstruction quality, as our feature encoder was not designed to capture fine details. Additionally, we utilized SLAM to create a more accurate 2DCLIP Map. Our evaluation involved constructing the map on top of the results from Segment Anything Model (SAM) [4] and conducting tests on NeRF Synthetic Mic data. We validated our results in hardware with a Franka Panda manipulator.

Dataset

We utilized ToyBox13 [12] to evaluate the performance of our model. This dataset includes synthetic scenes containing

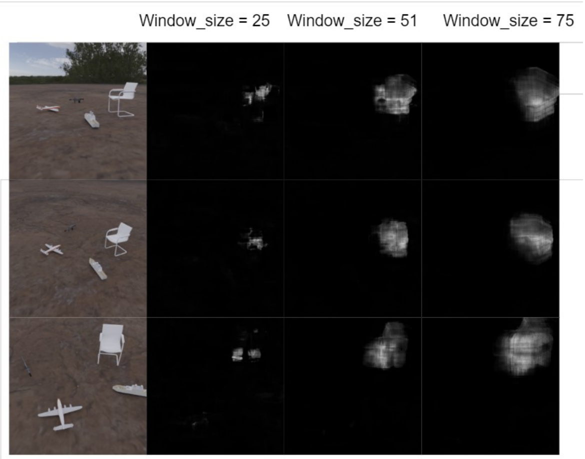

Figure 3. CLIP feature encoding on different window sizes.

various 3D objects that are realistically rendered using a path tracer. The scenes are described by posed frames, each of which includes an RGB image, a semantic map, and a depth map captured from a shared camera position. The dataset's realistic rendering and diverse object types align with our goal of reconstructing real-world scenes with semantic understanding for robotic grasping.

Window Size Tests

Based on our observations, window sizes ranging from 30 to 50 seem to produce the most desirable shape representations. However, we recognize that the ideal window size may vary depending on the size of the objects in the scene. To illustrate this, we have included Figure 3, which shows the result of querying the word "chair" on the encoded feature map frames in different sizes.

Novel View Synthesis

Following training, we generate novel views of the feature maps and query texts from these maps. As depicted in Figure 4, we find that the regions containing the relevant objects receive higher scores than other areas (top middle, bottom left). We observed that the geometries of larger objects, such as chairs, are better captured, whereas the geometries of smaller objects, such as airplanes, tend to be blurrier. This may be attributed to the window sizes used to encode the CLIP feature, as well as the reconstruction quality of the RGB network.

Object grounding

The CLIP-feature field enables precise 3D object grounding, providing access to geometry and accurate object coordinates. To assess the model's capability, we conducted several object grounding tests using various text prompts.

Grounding Objects by Attributes

We investigated grounding objects by their attributes, such as "white," as depicted in Figure 4, bottom middle. We observed that every white object in the scene had a higher similarity score in their position, demonstrating the model's ability to ground objects based on their attributes. Additionally, in Figure 4, top right, we show that when we query one object next to another, both objects have high similarity scores to the text prompt.

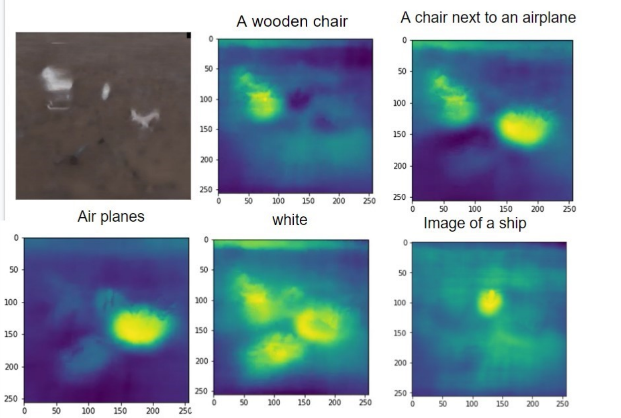

Figure 4. Output from querying different text prompts in the reconstructed CILP feature field.

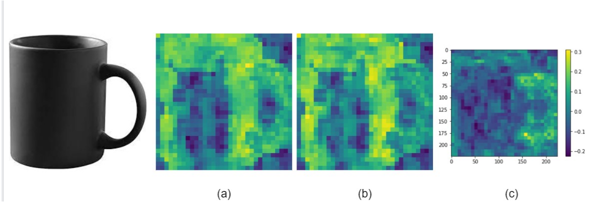

Figure 5. Output from grounding parts on an object (a) heatmap of query: a cup with a handle, (b) heatmap of query: a cup, (c) subtraction result of (a)-(b)

Grounding Components on an Object

To fully leverage the fine geometry accessible for robotic grasping tasks, we aimed to determine whether the model can ground specific parts of an object we want to grasp, such as the handle of a cup or the neck of a bottle. We demonstrate that the model can achieve this by calculating the difference between similarity scores for two text prompts.

Figure 5 shows our approach, where we queried "a cup with a handle" (a) and "a cup" (b). Using only the query result from (a), we were unable to locate the handle, as the entire cup and the handle had similarly high similarity scores. However, by calculating the difference between (a) and (b), we observed that the handle had a relatively higher score than the rest of the cup parts, enabling us to identify the precise location of the handle.

Robotic Grasp

The implemented code runs on ROS1 and is demonstrated on a Franka Panda manipulator, as depicted in Figure 6. Currently, the object position is hard-coded in the simulated environment, and a simulated BRICS environment is also created to prevent collisions. To enable communication between BRICS and the robotic arm, we transform the detected region in NeRF space to one of the BRICS camera's coordinates (Figure 7). The robotic arm's position is also transformed to that fixed camera coordinate. With this unified coordinate system, the grasp

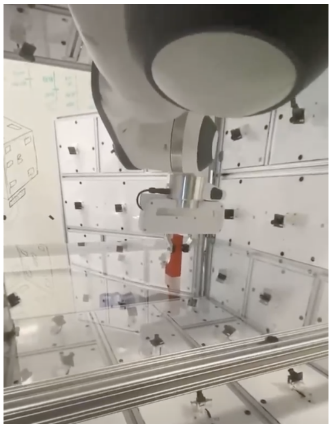

Figure 6. Demonstration of the implemented code on a Franka Panda manipulator in a simulated environment.

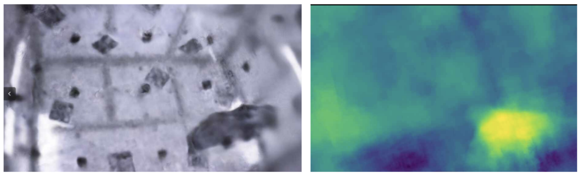

Figure 7. NeRF space transformed to BRICS camera coordinates.

point detection and motion plan algorithm can be executed seamlessly.

Conclusion

We present a CLIP-NeRF approach, a novel framework for learning 3D semantic scene representations using CLIP and NeRF. We demonstrated that our framework can be trained with weakly-supervised web-image trained models and can perform simple text query tasks. We also showed that the RGB density can guide the extraction of structure-preserved CLIP-encoded objects in 3D space, making it useful for generalizing robotic grasping tasks.

There are some limitations that could be addressed in future research. First, the accuracy of the model can be improved by fine-tuning the five hyper-parameters for different cases to achieve the best performance. Second, the inference speed can be improved, as it currently takes about 12 seconds to infer. Third, the current approach requires the object to be put in the BRICS Box to construct the NeRF, which limits its use in open-world scenarios.

There are also several areas that require further research. A method to extract grasp points from the rendered saliency map of the 3D CLIP Neural Radiance and estimate the grasp point in the NeRF space, building on related work such as GraspNeRF [1], could be implemented. Expanding to verbal language guided tasks could also generalize the approach to a wider range of objects and scenes.

We believe that by addressing these limitations and future research directions, our approach has the potential to advance the field of robotic grasping and enable robots to perform more complex tasks in real-world scenarios.

ACKNOWLEDGMENTS

We thank Tuluhan and Benedict for help with the Franka setup.

REFERENCES

[1] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. 2022. GraspNeRF: Multiview-based 6-DoF Grasp Detection for Transparent and Specular Objects Using Generalizable NeRF. *arXiv preprint arXiv:2210.06575* (2022).

[2] Huy Ha and Shuran Song. 2022. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *Conference on Robot Learning*.

[3] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: Language Embedded Radiance Fields. *arXiv preprint arXiv:2303.09553* (2023).

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and others. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[5] Jiahao Li, Greg Shakhnarovich, and Raymond A Yeh. 2022. Adapting clip for phrase localization without further training. *arXiv preprint arXiv:2204.03647* (2022).

[6] Antonio Morales, Beatriz Leon, Eris Chinellato, and Raúl Suárez. 2022. Current Challenges and Future Developments in Robot Grasping. *Frontiers in Robotics and AI* (2022), 194.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[8] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*. PMLR, 894–906.

[9] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. 2021. iMAP: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6229–6238.

[10] Yu Sun, Joe Falco, Máximo A Roa, and Berk Calli. 2021. Research challenges and progress in robotic grasping and manipulation competitions. *IEEE robotics and automation letters* 7, 2 (2021), 874–881.

[11] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. 2022. Language grounding with 3d objects. In *Conference on Robot Learning*. PMLR, 1691–1701.

[12] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. 2021. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260* (2021).

[13] Chong Zhou, Chen Change Loy, and Bo Dai. 2021. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071* (2021).